See 1473

COMPUTER SCIENCE TECHNICAL REPORT SERIES

UNIVERSITY OF MARYLAND
1807-1856-1920

UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND
20742

TR-539 May 1977
DAAG53-76C-0138

REGION TRACKING USING DYNAMIC PROGRAMMING

David L. Milgram
Computer Science Center
University of Maryland
College Park, MD 20742

ABSTRACT

The objects contained in a sequence of images may be tracked from frame to frame by defining a comparison function which evaluates the difference between descriptions of object regions in adjacent frames. One can then apply dynamic programming to discover the most temporally consistent object region. Removing all descriptions of this region from all frames allows dynamic programming to be reapplied iteratively.

The support of the U. S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (ARPA Order 3206) is gratefully acknowledged, as is the help of Mrs. Shelly Rowe.

1. Introduction

As imaging hardware capable of capturing real-time data evolves, the need increases for software which tracks the motions of object regions. An extensive literature already exists on tracking and motion. For a recent survey, with a good bibliography see Martin and Aggarwal [1].

In this paper, we have devised a simple region tracking scheme which uses dynamic programming to organize the search for consistent descriptions of regions appearing in the frames of the sequence. The algorithm works in conjunction with a region proposer called "Superslice", described in [2]. We present a brief description of Superslice for clarity.

2. Superslice - the region proposing algorithm

If one assumes that the desired objects in a scene may be extracted by thresholding at some set of gray levels, one may view the extraction of the above-threshold connected components as a process for producing candidate object regions. One may then classify the candidates into object regions and accidents (noise regions produced by thresholding). The Superslice algorithm uses two general heuristics and one piece of user-supplied knowledge. The first heuristic demands that the interior of a region contrast significantly with its surround. The second heuristic requires that the border points of the region correspond to positions of maximal edge detector response. Measures associated with these heuristics may be computed as the connected components are extracted. In addition, the user may control the false alarm rate by specifying a size range for object regions. The two measures and the size range are then used to build a classification.

The regions which survive the classification process have an inherent forest-like structure. Since an object may be extracted by thresholding over a range of adjacent gray levels, the candidate regions corresponding to the object can be ordered by containment. The containment relation defines the forest-like structure. A sequence of nested regions which do not differ much in size and shape may be considered to be a set of "exemplars" of the object. Not all regions which survive the classification step correspond to objects, however. A certain number of accidents tend to be present as well. All regions which do survive will be called "candidate object regions".

Other statistics besides the contrast, edge coincidence and size measures are computed during the analysis. These may include texture, shape, and positional information. The frame to frame tracking process will use these features to build consistent temporal sequences of candidate object regions.

3. Evaluating candidate object regions

There are two issues involved in finding a best sequence of exemplars for an object by choosing one exemplar per frame. First, within each frame we wish to select the "best" among all exemplars for each object. Second, on a frame to frame basis, we wish to avoid sudden changes in size, shape, position or other descriptive features associated with the tracked object. Realizing the former goal involves defining a figure of merit so that all exemplars of the same object may be compared among themselves. The Superslice procedure provides such a figure of merit based on the object/accident discriminant. Other things being equal, one would wish to choose the exemplar which represents the underlying object most closely. In the absence of specific models for particular object types, the general requirements of good contrast and good border/edge match are appropriate. In the example to be presented, the figure of merit was a weighted sum of the three features, the third being the number of edge points internal to the region.

Consider a sequence of exemplars for a single object corresponding to a range of thresholds. Because the sequence is nested we may speak of the "smallest" exemplar, etc. Assume that a "correct" exemplar is known (say, from additional ground truth). A "too small" exemplar will tend to have lower contrast (since the exterior neighbors of the border cells will in fact lie within the object region) and lower border/edge match (since the border points lie behind or just at the shoulder of the edge ramp, while the maximum edge response lies along the middle of the edge ramp). A "too big" exemplar will exhibit a similar response pattern. However, for the exemplars closest to the correct one, the features will not behave consistently. The exemplars just larger than the correct one often have more contrast due to the (higher) ratio of interior points to border points.

Given a figure of merit, one could choose the best exemplar from each sequence in the forest. The correspondence of exemplars from frame to frame might then be made by some simple matching procedure or by some modification of the procedure to be described below. Tracking based on best exemplars runs the risk that the best exemplar of an object in one frame bears little resemblance to its best exemplar in the next. This may be due to noise which afflicts certain frames more than others. The premise here is that an object does not change character significantly from frame to frame and that the changes which do occur should be smooth rather than abrupt.

Associated with each candidate object region is a feature vector (along with the figure of merit). We can measure the disparity or inconsistency between the candidate object regions by computing the normalized Euclidean distance between the two feature vectors. By using several features to define a disparity measure, we reduce the sensitivity of the method to gross changes with respect to a single feature. As with the figure of merit, we may weight the features entering the disparity computation according to their frame to frame consistency. This weighting can be guided by the semantics of motion (e.g., in plastic deformation, area and perimeter will remain roughly constant, but second order moments will change).

For the example we investigated, an equal weighting of features was chosen.

4. The dynamic programming model

In the previous section, we discussed two evaluation functions: a static evaluation function $S(c)$ defined for each candidate object region c based on the figure of merit, and a dynamic evaluation function $D(c,c')$ which for any pair of candidates defines their disparity. Let $c_1,\ldots,c_N = c$ be a sequence of candidate object regions, ending with the region c. We define the total cost of the region c as $T(c) = \sum_{i=1}^{N} S(c_i) + \sum_{i=1}^{N-1} D(c_i, c_{i+1})$. $S(c)$ is defined so that a perfect exemplar has a score of 0. Similarly $D(c,c) = 0$.

Let $\{c_{ij};\ j = 1,\ldots,N_i\}$ be the set of candidate regions in the ith frame, $i = 1,\ldots,M$. We define the dynamic programming problem as: find $\{c_{i\pi_i};\ i = 1,M\}$ such that $T(c_{M\pi_M})$ is minimum over all selection functions, π. The solution is achieved by the following:

Basis step: $T(c_{1j}) = S(c_{1j});\ j = 1,\ldots,N_i$

Iterative step:
$$T(c_{i+1j}) = S(c_{i+1j}) + \min_{K=1,\ldots,N_i} \{T(c_{ik}) + D(c_{ik}, c_{i+1j})\}$$
$$\text{for } j = 1,\ldots,N_{i+1}$$

The above procedure finds the minimum cost sequence of candidate object regions. Candidate regions which are accidental are unlikely to persist from frame to frame; thus their D terms are likely to be large, thereby increasing the total cost of any sequence which includes them. Note that there will be many sequences which are only slightly more costly than the minimum. These suboptimal sequences will be based on other exemplars for the same object. The optimal sequence is thus optimal for the particular formulations of S and D. Giving more weight to S and less to D will tend to select best exemplars; while the reverse weighting will tend to favor frame to frame consistency. Once again, a semantic model can provide guidance.

In general, the image sequence may contain more than one object. The scheme described above identifies the "best" object region sequence. In order to extract region sequences corresponding to other objects in the image sequence, we must delete all candidate object regions accounted for by the optimal sequence. The inherent data structure specifies which regions are exemplars for each object. By deleting all candidate object regions in each frame which are similar to the selected region of the optimal sequence (i.e., contain it or are contained in it), we can set the stage for another application of dynamic programming. This process is repeated until only very poor (high cost) sequences are obtained. Presumably, at this point all objects have been accounted for.

Occasionally, a deletion step may leave a particular frame empty of candidate object regions. This may occur for two reasons: All objects were accounted for by the last dynamic programming step, or the candidate region proposer failed to elicit an exemplar for an actual object.

In the former case, the process will have terminated. The latter case can be handled by associating a fixed "empty frame" cost which is the price paid for skipping a frame. Of course, one can't know which case applies. The conservative approach is always to assume the second case and apply the empty frame cost. The termination criterion will then be based on a threshold for the total cost, i.e., terminate when only costly sequences remain.

The problem of an object leaving the field of view can be handled in a different manner by flagging candidate object regions which lie on the border of the image. A partial sequence whose last element is flagged but which overall has low cost can be accepted as depicting an object which has moved off the image.

5. Experimental results

The dynamic programming algorithm described above has been implemented and tested on a sequence of ten windows of FLIR data containing a tank (Figure 1). These windows have already been smoothed by a 3x3 median filter to provide better response to thresholding. The Superslice algorithm extracted a modest number of candidate object regions. Figure 2 displays these regions (although for nested sequences only the best static exemplar is displayed). Table 1 shows the feature values associated with each candidate in the first two frames. The solution to the dynamic programming problem was computed and the exemplars which correspond to the solution are shown in Figure 3. There are of course many suboptimal solutions which are quite similar to this one. Their cost is not significantly greater than the minimal cost. When the indicated regions were deleted along with all other similar candidates, the only remaining regions corresponded to noise and any minimal cost path attempting to span several frames was substantially more costly than the optimal path or any of its similar suboptimal paths. It seems reasonable to establish thresholds for static and dynamic cost in order to prune the search space. More sequential data bases are needed to determine the extent to which these comments are valid.

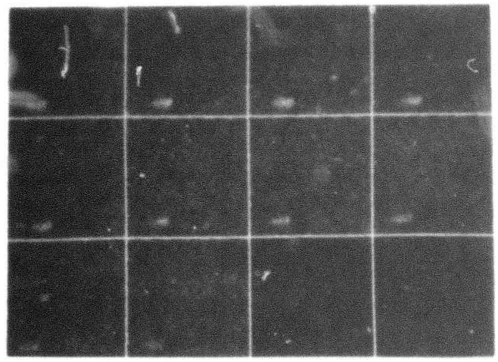

Figure 1. A sequence of 10 median filtered FLIR windows of a tank.

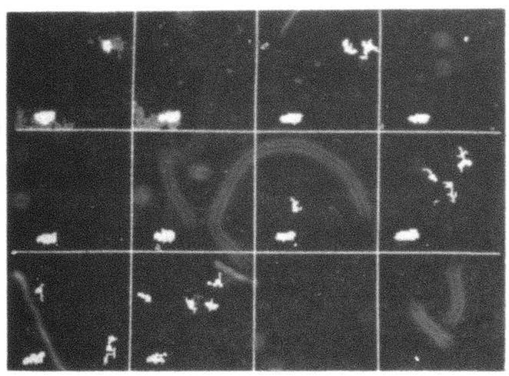

Figure 2. Output of the Superslice region proposing algorithm.

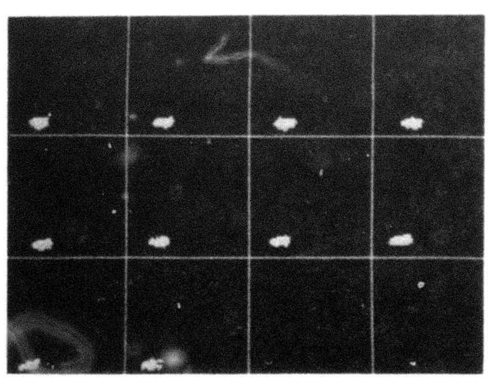

Figure 3. Optimal sequenced regions using dynamic programming.

Frame-component #	Thresholded at	Area in points	Contrast	% Edge match	# Interior edge points	Static evaluation
1-1	25	53	6.3	50	1	.53
1-2	24	53	6.3	50	1	.53
1-3	23	57	6.5	53	3	.50
1-4	22	61	7.3	53	4	.49
1-5	21	67	7.2	47	7	.51
1-6	20	76	7.7	42	7	.52
1-7	19	87	8.5	36	8	.51
1-8*	19	21	.7	72	0	.50
1-9	18	107	5.6	37	9	.57
1-10*	18	23	.7	82	0	.45
1-11	17	126	6.4	33	9	.55
1-12*	17	33	1.4	73	0	.45
1-13*	16	199	3.1	26	12	.66
1-14*	16	60	1.2	57	0	.54
1-15*	15	243	3.1	30	15	.63
1-16*	15	99	1.8	59	1	.48
2-1	25	49	6.2	73	1	.42
2-2	24	51	6.1	71	1	.43
2-3	23	55	6.8	73	1	.40
2-4	22	58	7.4	66	2	.42
2-5	21	63	7.5	54	5	.48
2-6	20	65	7.5	50	6	.50
2-7	19	71	8.1	47	9	.49
2-8	18	99	10.3	33	11	.47
2-9*	17	193	3.6	24	13	.66
2-10*	16	229	1.7	28	13	.70
2-11*	15	248	1.8	31	13	.68

Table 1a. Feature values for components in frames 1 and 2 which enter into the static evaluation. "*" means that the component is an accident.

Frame-component #	Height	Width	Perimeter	Centroid	2nd Central Moments X^2	Y^2	XY
1-1	7	11	34	17,57	124.	414.	226.
1-2	7	11	34	17,57	124.	414.	226.
1-3	7	11	32	17,57	128.	429.	235.
1-4	7	11	32	17,58	133.	445.	244.
1-5	8	12	34	17,58	139.	468.	255.
1-6	8	13	38	17,58	148.	500.	272.
1-7	10	13	45	17,58	155.	539.	289.
1-8*	6	5	18	50,20	222.	90.	142.
1-9	11	14	46	16,59	169.	603.	320.
1-10*	6	5	17	50,20	233.	95.	148.
1-11	11	15	51	17,59	190.	657.	353.
1-12*	7	9	26	50,20	280.	115.	179.
1-13*	12	26	93	13,59	185.	833.	393.
1-14*	7	13	44	51,20	392.	153.	245.
1-15*	13	33	107	15,59	234.	925.	465.
1-16*	14	15	65	53,20	521.	196.	320.
2-1	6	12	33	20,57	136.	394.	232.
2-2	6	12	31	20,57	139.	402.	236.
2-3	7	12	33	20,57	143.	418.	245.
2-4	7	13	35	20,57	148.	430.	252.
2-5	7	13	37	20,57	153.	450.	263.
2-6	7	13	38	19,57	155.	458.	267.
2-7	7	13	38	19,57	162.	478.	279.
2-8	11	16	51	19,58	189.	573.	329.
2-9*	15	25	109	13,58	185.	810.	387.
2-10*	16	25	102	13,59	191.	883.	411.
2-11*	17	25	86	13,58	196.	918.	425.

Table 1b. Additional feature values for components in frames 1 and 2 which enter into the disparity computation.

6. Conclusion

Objects may be tracked in a sequence of scenes in which frame to frame change is slight. The dynamic programming method relies on the heuristic that even though some motion or change may have taken place in the scene, descriptions of the same object tend to cluster more closely than do descriptions of different objects. Thus, a measure based on similarity and consistency can provide a reliable match function even in a dynamic environment. After an object has been tracked consistently through a sequence of frames, one may measure its motion, deformation, etc.

Bibliography

1. W. M. Martin and J. K. Aggarwal, Dynamic scene analysis: a survey, to appear in Computer Graphics and Image Processing, 1977.

2. D. L. Milgram, Region extraction using convergent evidence, Proceedings: Image Understanding Workshop, Minneapolis, MN, April 1977.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER (9)
4. TITLE (and Subtitle) (6) REGION TRACKING USING DYNAMIC PROGRAMMING.		5. TYPE OF REPORT & PERIOD COVERED Technical Rept.,
		6. PERFORMING ORG. REPORT NUMBER (14) TR-539
7. AUTHOR(s) (10) David L. Milgram		8. CONTRACT OR GRANT NUMBER(s) (15) DAAG-53-76C-0138,
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Science Ctr. Univ. of Maryland College Park, MD 20742		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order-3206
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Night Vision Lab. Ft. Belvoir, VA 22060		12. REPORT DATE (11) May 1977
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 19p.		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Tracking
Change detection
Dynamic programming
Image processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The objects contained in a sequence of images may be tracked from frame to frame by defining a comparison function which evaluates the difference between descriptions of object regions in adjacent frames. One can then apply dynamic programming to discover the most temporally consistent object region. Removing all descriptions of this region from frame to frame allows dynamic programming to be reapplied iteratively.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

403 018

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)